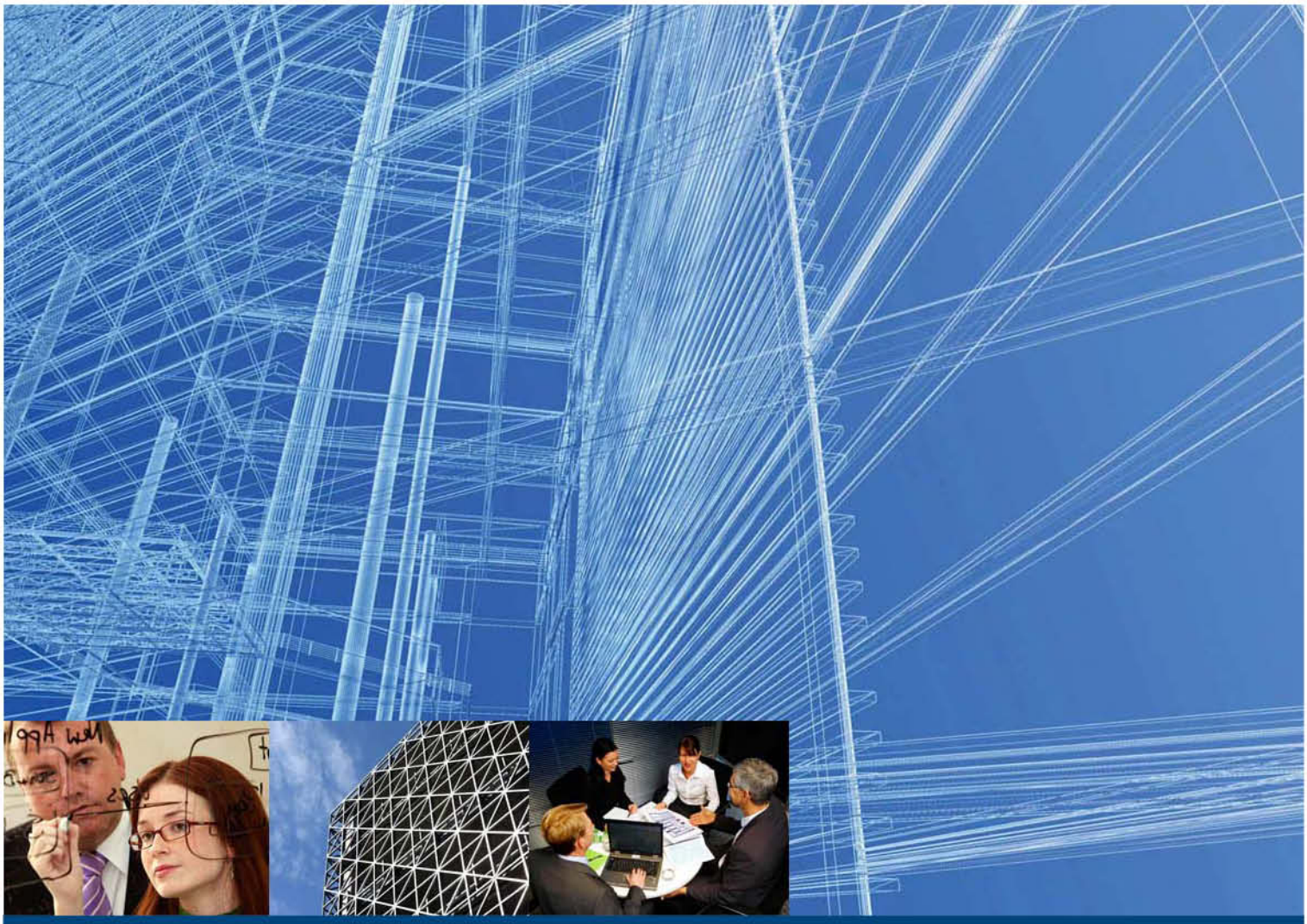

Web application security testing guideline

Final

December 2011

v1.0.0

PUBLIC

Queensland Government Enterprise Architecture

Queensland Government Chief Technology Office
Department of Public Works

Queensland Government

Document details

Security classification	PUBLIC		
Date of review of security classification	December 2011		
Authority	Queensland Government Chief Information Officer		
Author	Queensland Government Chief Technology Office		
Documentation status	Working draft	Consultation release	☑ Final version

Contact for enquiries and proposed changes

All enquiries regarding this document should be directed in the first instance to:

Executive Director
Queensland Government Chief Technology Office
qgcto@qld.gov.au

Acknowledgements

This version of the *Web application security testing guideline* was developed and updated by the Queensland Government Chief Technology Office.

Feedback was also received from a number of staff from various agencies, which was greatly appreciated.

Copyright

Licence

Information security

This document has been security classified using the Queensland Government Information Security Classification Framework (QGISCF) as PUBLIC and will be managed according to the requirements of the QGISCF.

Table of contents

1 Introduction

1.1 Purpose

The *Web application security testing guideline* is structured to help agencies increase their assurance the web applications they use have been developed in a manner that ensures the confidentiality, integrity and availability of the agency data within these applications is maintained in accordance with the agency risk profile. This guideline is intended to maintain consistency with, and support for, Information Standard 18: Information Security (IS18).

A Queensland Government Enterprise Architecture guideline is non-mandatory and provides information for Queensland Government agencies on the recommended practices for a given topic area.

1.2 Audience

This document is primarily intended for:

- agency staff and operational areas involved in web application development, acquisition (including code developed for an agency by third party/outsourced providers) and maintenance services
- agency information security management
- information security governance bodies.

1.3 Scope

This guideline provides recommended practices for web application security testing and is product/vendor independent.

2 Background

2.1 Why was this guideline developed?

The Queensland Audit Office's Information Security Governance Review 2010-11 detailed a concern that:

> *web application reviews appear to have been subjected to different tests when the requested scope from the clients has been similar. Audit has also noted varying quality of reports, such as where the testing identified weaknesses, but where the information was omitted in the test report.*

The Queensland Audit Office review went on to observe that the Open Web Application Security Project's (OWASP) Application Security Verification Standard (ASVS) was 'the internet community's de facto standard for web application testing'.

This guideline has been developed to assist agencies to establish and implement an approach to web application security testing, using the OWASP ASVS, which will ensure consistent results that allow incremental measurement over time. This will facilitate both agency security maturity advancement and inter-agency learning opportunities.

2.2 Relationship to other QGEA documents

The *Web application security testing guideline* complies with the implementation of IS18. While relevant to the majority of the principles of IS18, it aligns specifically to the principles relating to:

- access management
- system acquisition, development and maintenance (including third party/outsourced services such as application development).

The suite of supporting documents to assist agencies in meeting their web application security assurance requirements is shown in the following diagram (figure 1 below).

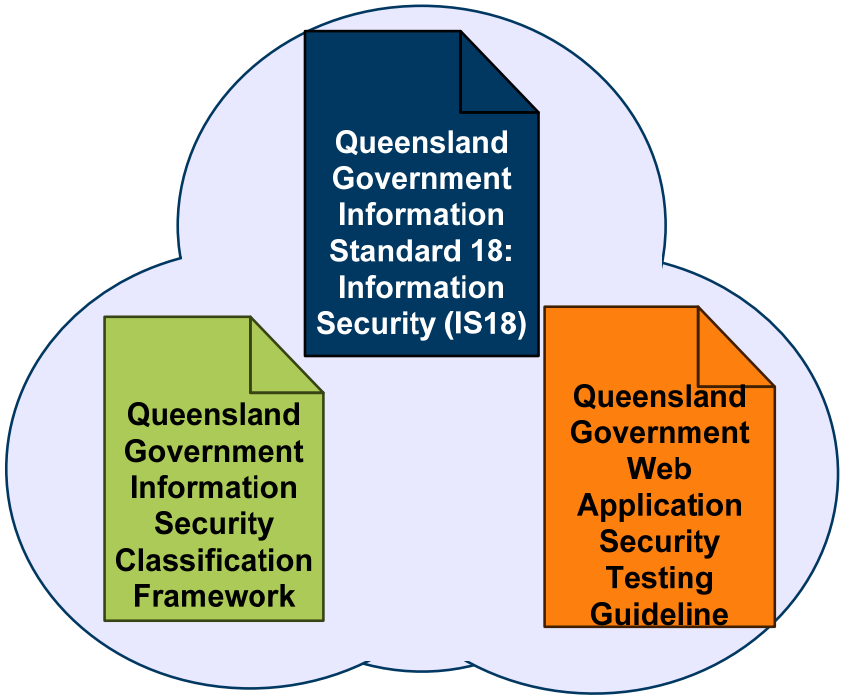

Figure 1: Web application security testing – suite of documents

3 OWASP ASVS

Formed in 2001, the OWASP is a not for profit international open community that produces frameworks, guidance and tools to enable organisations to ensure that the applications they develop, purchase, operate, and maintain can be trusted.

One of their sub projects is the ASVS which details an approach to web application testing that, when applied, ensures agencies can gain assurance the software they build and buy is compliant with a specific pre-defined security level as it will have been verified according to a common standard.

ASVS achieves this through a hierarchical process, displaying increasing rigor as the criticality of the system (known as the target of verification or TOV) increases.

Each level of verification builds upon the previous level/s. A summary of each level is included below:

3.1 Level 1 testing

For applications where some confidence in the correct use of security controls is required, level 1 ('automated verification') is typically appropriate. Threats to security will typically be viruses and worms and these threats are tested using automated tools (for example NESSUS) augmented with manual verification. This level only provides partial application security verification coverage. The manual verification is not intended to make the application security verification performed at this level complete, only to verify that each automated finding is correct and not a false positive. This level of verification would be best suited to agency systems handling, as a maximum, **PUBLIC** data or **UNCLASSIFIED** data.

3.2 Level 2 testing

For applications that handle personal transactions, conduct business-to-business transactions, process credit card information, or process personally identifiable information or where some confidence in the correct use of security controls and confidence that the security controls are working correctly is required, level 2 ('manual verification') is typically appropriate. Threats to security will typically be viruses, worms and unsophisticated opportunists such as attackers with professional or open source attack tools. The scope of verification includes all code developed or modified for the application as well as examining the security of all third party components that provide security functionality for the application. This level of verification would be best suited to agency systems handling, as a maximum, **X-IN-CONFIDENCE** data.

3.3 Level 3 testing

For applications that handle significant business-to-business transactions, including those that process healthcare information, implement business-critical or sensitive functions or process other sensitive assets, level 3 ('design verification') is typically appropriate. Threats to security will typically be viruses and worms, opportunists and possibly determined attackers (skilled and motivated attackers focusing on specific targets using tools including purpose-built scanning tools). The scope of verification includes all code developed or modified for the application, as well as examining the security of all third party components that provide security functionality for the application. Level 3 ensures that security controls themselves are working correctly, and that security controls are used everywhere within the application they need to be used to enforce application-specific policies. This level of verification would be best suited to agency systems handling, as a maximum, **PROTECTED** data.

3.4 Level 4 testing

For critical applications that protect life and safety, critical infrastructure, or defence functions, level 4 ('internal verification') is typically appropriate. Level 4 ensures that security controls themselves are working correctly, that security controls are used everywhere within the application they need to be used to enforce application-specific policies and that secure coding practices were followed. Threats to security will be from determined attackers (skilled and motivated attackers focusing on specific targets using tools including purpose-built scanning tools). The scope of verification expands beyond the scope of level 3 to include all code used by the application. This level of verification would be best suited to agency systems handling, as a maximum, **HIGHLY PROTECTED** data.

Caveats

1. Where an agency cannot have a system or application actively tested (if, for example, it is commercial off-the-shelf shrink-wrap software) the agency should seek independent certification from the vendor that attests to the ASVS level applicable to the target of verification. If the target of verification has not been assessed to the ASVS standard then the vendor should provide:
   a. contractual assurance that their solution meets all requirements of the applicable ASVS level
   b. contractual liability to the agency should the target of verification later prove to have insufficient controls.
2. If the agency utilises a commercial off-the-shelf shrink-wrap software product but modifies it through custom development (add on modules or functionality etc) either in-house or outsourced, the agency should have the resulting solution tested as per this guideline.
3. The testing detailed in the ASVS provides the minimum standard that should be applied. Testers should not feel constrained to stay within the boundaries of the tests specified and should use any appropriate tests (within the constraints provided by the agency) to ensure vulnerabilities are discovered and rated.

Assumptions

Testers should be skilled in application security testing. The agency should ensure that testers have appropriate skills to perform the work. This may include a recognised certification from an industry body. Additionally, testers should seek input on vulnerabilities from multiple sources, such as the CWE/SANS top 25 most dangerous software errors and OWASP top 10 vulnerabilities.

4 Process

To ensure that agency's specify web application security testing to a level of rigor appropriate to the data contained within the system in question (the target of verification), the agency should classify the data.

Information is an asset of the agency and the government has established accepted levels of behaviours when handling certain types of information. As not all information is of equal value, and because resources are limited, it is important to prioritise the data and systems that need protection. Using the Queensland Government Information Security Classification Framework, agencies should define the classification for the data within the target of verification.

By defining the classification of the data within the target of verification, agencies:

- gain an understanding of the sensitivity and criticality of the data
- comply with the mandatory requirements of IS18
- document the requirements.

Agencies have the option to use internal resources to undertake the testing or, if appropriate internal resources do not exist, engage external service providers. With either option, the agency should document their requirements into an internal scope of work or a request for offer (RFO) for security testing using the most appropriate level of assessment in the ASVS. Agencies may take guidance as to how to best align the requirements of the data with the rigor of testing needed using section 3 (above), however agencies should always be cautious and mediate upwards when selecting a level of testing. This is

especially important when one considers the potential impact to an agency resulting from the magnitude of the potential loss (i.e. the loss of one credit card number may have minimal impact on an agency compared to the impact from the loss of 10,000 credit card numbers).

The sponsor should seek assistance from their agency legal/contracts group, however some initial guidance on how to negotiate and capture important contractual terms and conditions related to the security of the software to be developed or tested may be found in the OWASP Legal Project.

Agencies should ensure their RFO specifies testing providers produce reports that comply with the ASVS standard so as to facilitate consistent ongoing assessment. As part of the justification component of their report, providers should ensure that they make an assessment of the risk posed to the agency from discovered vulnerabilities, using the risk matrix in table 1 (below). Testers may also include industry accepted scores, such as the Common Vulnerability Scoring System, for additional detail.

Likelihood	Severity				
	None/negligible	Minor	Moderate	Major/high	Severe/ very high
Likely	Medium	Medium	High	High	High
Possible	Low	Medium	Medium	High	High
Unlikely	Low	Low	Medium	High	High

Table 1: Risk matrix

In calculating this assessment of risk, the provider should estimate the likelihood of a vulnerability being exploited by an attacker, using table 2 (below), and the potential severity to the agency, using the information security incident severity matrix provided in the Information security incident category guideline. This view should consider and specify:

- the skills that would be required by an attacker to exploit the vulnerability
- any existing controls
- residual risk after any recommended remedial controls are implemented.

Likelihood category	Description
Likely	Can be expected to occur at least once in a 12-month period or there is a greater than 50 per cent chance of this risk eventuating.
Possible	Could occur at least once in a 36-month period or there is a 10 to 50 per cent chance of this risk eventuating.
Unlikely	Conceivable but highly unlikely to occur more than once in 5 years or there is a less than 10 per cent chance of this risk eventuating.

Table 2: Likelihood of event

5 Secure development frameworks

When information security is not considered and incorporated at all phases of a development cycle the end result is often a solution with inadequate controls in place that then either requires expensive rework or fails to meet production requirements.

The National Institute of Standards and Technology estimates that any code remediation performed after production implementation results in up to 30 times the cost of remedial action performed during the design phase. The graph below shows that the cost for remediating vulnerabilities is highest after an application has been deployed. It includes not only the costs involved with re-engineering a system to mediate vulnerability but also the cost of lost user productivity.

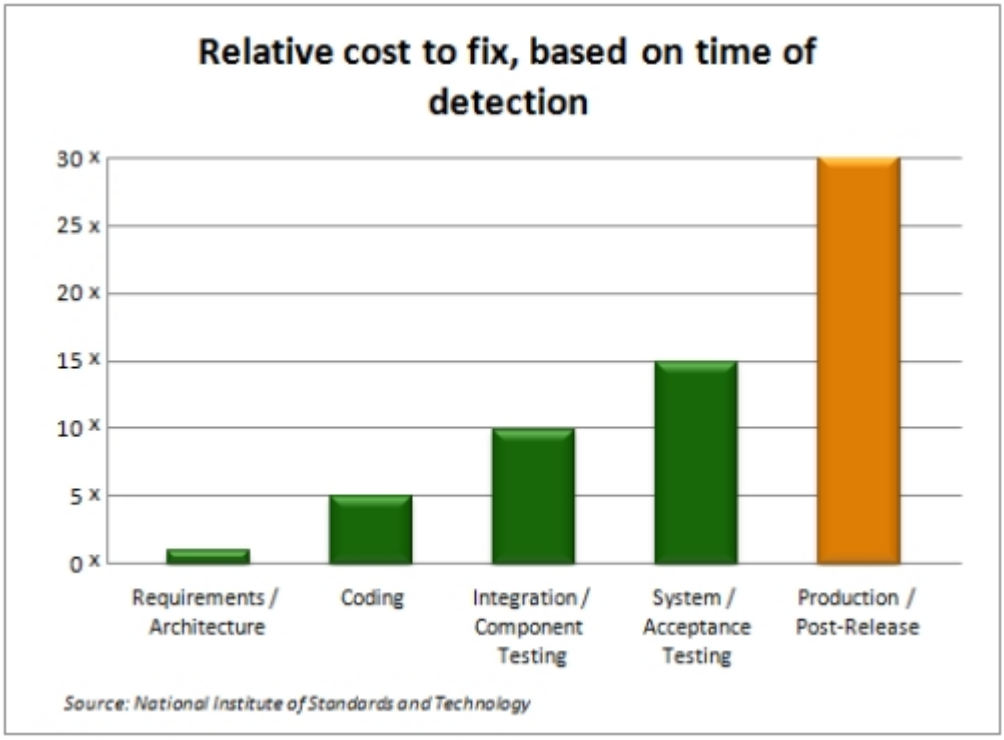

Figure 2: Cost for remediating vulnerabilities

Where the results from testing highlight deficiencies in web application security, it is recommended that agencies review their development and acquisition procedures to ensure that their processes are consistent with secure development practices.

By following a defined process, such as those listed below that systematically address software security during the development phase, vulnerabilities are more likely to be found and fixed prior to application deployment, thereby reducing the total cost of software development.

- Software Assurance Maturity Model – an open source framework to help organisations formulate and implement a strategy for software security that is tailored to the specific risks facing the organisation. The resources provided by the Software Assurance Maturity Model will aid in:
  - evaluating an organisation's existing software security practices
  - building a balanced software security assurance program in well-defined iterations
  - demonstrating concrete improvements to a security assurance program

  - defining and measuring security-related activities throughout an organisation, additionally, it is recommended that agencies ensure that vendors who produce either custom-built solutions for the agency or commercial off-the-shelf solutions also incorporate similar assurance steps into their development frameworks.
- Comprehensive, Lightweight Application Security Process – provides a structured process for integrating security assurance points into the software development lifecycle.
- Microsoft® Security Development Lifecycle – a free framework provided by Microsoft that aims to reduce the number and severity of vulnerabilities in software through a collection of security practices, grouped by the phases of the traditional software development life cycle.

Where security vulnerabilities are found in existing code, agencies may find some solutions available through the OWASP Enterprise Security API, an open source web application security control library designed to streamline the retrofitting of security controls into pre-existing code. These libraries also serve as a solid foundation for new applications development initiatives.

6 References

- Common Vulnerability Scoring System
- Information security incident category guideline
- Information Standard 18: Information Security (IS18)
- Microsoft® Security Development Lifecycle
- Open Web Application Security Project (OWASP) Application Security Verification Standard
- OWASP Comprehensive, Lightweight Application Security Process
- OWASP Enterprise Security API
- OWASP Legal Project
- OWASP Software Assurance Maturity Model
- Penetration testing certifications – provides a reasonable list.
- Queensland Government information security classification framework